UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/046,496	10/29/2001	Carey Nachenberg	20423-05957	3384

34415 7590 05/18/2009

SYMANTEC/ FENWICK
SILICON VALLEY CENTER
801 CALIFORNIA STREET
MOUNTAIN VIEW, CA 94041

EXAMINER	
WILLIAMS, JEFFERY L	
ART UNIT	PAPER NUMBER
2437	

NOTIFICATION DATE	DELIVERY MODE
05/18/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptoc@fenwick.com
bhoffman@fenwick.com
aprice@fenwick.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/046,496
Filing Date: October 29, 2001
Appellant(s): NACHENBERG ET AL.

Brian Hoffman
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 2/13/09 appealing from the Office action mailed 7/10/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,721,721	Bates et al.	4-2004
7,099,916	Hericourt et al.	8-2006

Symantec Corporation, "Norton AntiVirus Corporate Edition", 1999, Version 1, pages 15, 22.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1 – 10, 12 – 17, 20, 22 – 32, and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bates et al. (Bates), U.S. Patent 6,721,721 B1 in view of Hericourt et al. (Hericourt), U.S. Patent 7,099,916.

Regarding claim 1, Bates et al. discloses:

entering a first computer virus status mode in response to a first computer virus outbreak report indicating a virus attack threat to a computer network (Bates et al., col. 1, lines 13-52). Bates et al. reports the outbreak of new and more sophisticated viruses, and in response, the system of Bates et al. is employed for the purpose of protecting against these outbreaks.

computing a first computer virus alert time corresponding to entry into the first computer virus status mode (Bates et al., fig. 7, elem. 214; col. 7, lines 20-35). Herein, Bates et al. discloses a method for accessing computer content on a local machine or on a network. Content is filtered based upon a generated virus alert time, a rule derived from relative time parameters (criterion) entered (via computer means, "computing") by a user in a virus status mode. The relative time parameters (i.e. "virus found in last 7 days", "not checked in last 14 days") are processed ("computing") into a rule, which is then utilized by the system to compare with the timestamps of an executable computer code and make determinations of trustworthiness (Bates et al., col. 11, lines 12-24; col. 13, lines 22-34; col. 17, lines 35-49; col. 18, lines 22-30).

comparing a time stamp of a executable computer code ... with the first computer virus alert time (Bates et al., col. 9, line 65 – col. 10, line 3; col. 11, lines 12-24; col. 12, lines 59-62; col. 13:23-34). Herein, Bates discloses that the virus alert time is compared to virus status information of an executable computer code, comprising a timestamp.

and determining the executability of the computer content in response to the result of the comparing step (Bates et al., col. 9, line 56 – col. 10, line 8; col. 11, lines 12-24). Bates et al. discloses that in response to a comparison, a determination of computer content executability is performed.

Bates discloses that a time stamp of the executable code corresponds, *inter alia*, to the time the code was virus scanned. However, Bates does not explicitly disclose that a time stamp of the executable computer code corresponds to an execution time of the computer code.

Hericourt teaches that virus scanning of executable code comprises an execution of the code (Hericourt, col. 3, lines 25-54).

It would have been obvious to one of ordinary skill in the art to recognize teachings of Hericourt within the system of Bates. This would have been obvious because one of ordinary skill in the art would have been motivated by the teachings of Hericourt that an effective virus scanning system should employ a scanning method comprising the execution of the code (e.g. Hericourt, col. 3, lines 51-61).

The examiner notes that the recited *"an earliest moment the computer code was allowed to execute on a computer coupled to the computer network"* must be given its broadest reasonable interpretation in harmony with the appellant's specification. Within the appellant's specification, the appellant reveals that this claimed timestamp of *"an earliest moment the computer code was allowed to execute"* represents a time when an executable file is virus checked by a networked computer and the resulting timestamp and hash value from the virus check is added to a database or memory table (e.g. see appellant's specification, par. 38, 50, 51, 60, 61; original claims 21 and 33).

Accordingly, the examiner notes that the prior art combination enables a *"computer coupled to the computer network"* (e.g. file hosting site [50], search engine [30] – see Bates, fig. 1; col. 8, lines 12-15, 43-48) to virus scan an executable, the resulting timestamp of virus scanned code corresponding to an "execution time" since virus scanning comprises execution of the executable (Hericourt, col. 3, lines 25-54). This scanning/execution of the executable by the networked computer causes the virus status information (e.g. the timestamp) to be added to a database or memory table (Bates, col. 8, lines 12-15, 43-48; col. 13, lines 24-34). Thus, the networked computer safely provides these virus-scanned, executable files as downloads, whereby they are subsequently executed at later times by thousands of users with client computers (e.g. see Bates, fig. 1: 20, 30, 50; col. 4, lines 35-55).

Thus, the combination enables:

... corresponding to an earliest moment the computer code was allowed to execute on a computer coupled to the computer network (Hericourt, col. 3, lines 25-54; Bates, fig. 1: 50, 30, 20; Bates col. 8, lines 12-15, 43-48).

Regarding claim 2, the combination enables:

receiving a first access control time based on the first virus outbreak report (Bates et al., fig. 7, elem. 214). The system of Bates et al. takes human input and "automatically" generates computer readable parameters.

and converting the first access control time into the first virus alert time (Bates et al., fig. 7, elem. 214; col. 12, lines 59-62). A "prior point in time" ("virus alert time") is derived from the period of time specified by element 214 ("access control time") and is compared to the timestamp of the file.

Regarding claim 3, the combination enables:

wherein the first access control time is a relative time stamp (Bates et al., fig. 7, elem. 214; col. 12, lines 59-62). A "prior point in time" ("virus alert time") is derived from the period of time specified by element 214 ("access control time") and is relative in time.

Regarding claim 4, the combination enables:

wherein the first access control time is a pre-determined time period for access control under the first computer virus status mode (Bates et al., fig. 7, elem. 214). The access control time is pre-determined by the user.

Regarding claim 5, the combination enables:

determining the presence of a value representing the computer content in a memory table of executable computer content (Bates et al., col. 7, lines 12-34).

Regarding claim 6, the combination enables:

wherein the computer content is not executed when the value representing the computer content is not present in the memory table of executable computer content (Bates et al., col. 11, lines 11-24; col. 3, lines 24-27). As disclosed by Bates et al., content not present in the memory table of executable computer content is flagged as untrustworthy. The invention as disclosed by Bates et al. is configurable to eliminate untrustworthy computer content from the list of accessible content, thus not providing access to the content for execution.

Regarding claim 7, the combination enables:

wherein the value is a hash value of the computer content (Bates et al., col. 12, lines 55-58).

Regarding claim 8, the combination enables:

wherein the computer content is determined to be executable only when the computer content is time stamped prior to the first computer virus alert time (Bates et al., col. 13, lines 42-59; col. 3, lines 24-27). Computer content that is time stamped prior to the first computer virus alert time is branded as trustworthy. Thus, the content would not be subjected to denial of access for execution.

Regarding claim 9, the combination enables:

entering types of computer codes that should be blocked from execution in response to the first computer virus outbreak report (Bates et al., col. 9, line 62 – col. 10, line 28);

and blocking execution of a computer code that belongs to the entered types of computer codes (Bates et al., col. 3, lines 24-27). The invention as disclosed by the combination is configurable to eliminate untrustworthy computer content from the list of accessible content, thus not providing access to the content for execution.

Regarding claim 10, the combination enables:

generating a second virus alert time in response to a second computer virus outbreak report; comparing the time stamp of the computer content with the second computer virus alert time; determining the executability of the computer content in response to the result of comparing the time stamp of the computer content with the second computer virus alert time *(Bates et al., col. 3, lines 5 – 15). The above limitations of claim 10 are essentially similar to claim 1 with the exception that they are directed to a second instance of the method of claim 1. The combination enables for the method of claim 1 produces a set of results. Thus, the combination enables a secondary instance of the method of claim 1, as a the word "set" dictates more than a singular occurrence of the method of claim 1.*

performing antivirus processing upon the computer content (Bates et al., col. 9, lines 62-66). The combination enables the processing of computer content for the likelihood of existing viruses.

Regarding claim 12, it is rejected, at least, for the same reasons as claim 1, and furthermore because the combination enables:

an access control console, for entering a first computer virus status mode in response to receiving a computer virus outbreak report indicating a virus attack threat to a computer network and for recovering a preselected virus access control time corresponding to said virus status mode (Bates et al., fig. 1, elem. 33; fig. 7);

an anti-virus module, coupled to the access control console, configured to compute a virus alert time based on the virus access control time and to compare a time *stamp of target computer code corresponding to an earliest moment the computer code was allowed to execute with the virus alert time prior to execution of the target computer content* (Bates et al., fig. 1, elem. 30; see rejections of claims 1 and 2).

and wherein the anti-virus module is further configured to determine the executability of the computer content in response to comparing the time stamp of the target computer content with the virus alert time (Bates et al., col. 9, line 56 – col. 10, line 8; col. 11, lines 12-24). The combination enables for in response to a comparison, a determination of computer content executability is performed. Thus the combination enables *content executability determination,* comprising an *anti-virus module*, used to determine the trustworthiness ("executability") of content.

Regarding claim 13, the combination enables:

a memory module for storing time stamps of the plurality of computer contents (Bates et al., fig. 1, elem. 46);

and an access control module, coupled to the access control console and to the memory module, for computing the virus alert time and for comparing the time stamp of each target computer content with the virus alert time (Bates et al., fig. 1, elem. 42; see rejections of claims 1 and 2).

Regarding claim 14, the combination enables:

a computer virus processing module, coupled to the access control module, for further processing a target computer content in order to determine the executability of the target computer content (Bates et al., fig. 1, elem. 44).

Regarding claim 15, the combination enables:

wherein the memory module stores a value representing each of the computer contents (Bates et al., col. 12, lines 52-65).

Regarding claim 16, the combination enables:

wherein the access control module is configured to determine the presence of the value in the memory module as representing a target computer content (Bates et al., fig. 3).

Regarding claim 17, the combination enables:

wherein the value is a hash value (Bates et al., col. 12, lines 52-65).

Regarding claim 20, it is rejected, at least, for the same reasons as claim 1, and furthermore because the combination enables:

creating a list of time-stamped executable computer contents (Bates et al., fig. 3, elem. 92).

entering a virus alert mode in response to a virus outbreak report indicating a virus attack threat to a computer network (Bates et al., fig. 2; col. 1, lines 13-52).

responsive to the virus alert mode, entering an access control message for specifying an access control rule for blocking the execution of suspicious or susceptible computer contents that have a time stamp corresponding to an earliest moment the computer file was allowed to execute, and the time-stamp is not before a computed virus alert time, the access control message including a first control parameter for computing the virus alert time (Bates et al., fig. 2; fig. 7; see rejections of claims 1 and 2).

receiving a request to execute a target computer content; and determining the executability of the target computer content based on the access control rule in the access control message (Bates et al., fig. 2).

Regarding claim 22, the combination enables:

receiving the access control message; automatically converting the first control parameter into the virus alert time; comparing the time stamp of the target computer content in the list with the virus alert time; and determining the executability of the target computer content based on the result of the comparing step (Bates et al., fig. 2, fig. 3, fig. 7; see rejections of claims 1 and 2).

Regarding claim 23, the combination enables:

applying an anti-virus operation upon the target computer content (Bates et al., fig. 3).

Regarding claim 24, the combination enables:

a second control parameter for specifying types of computer contents that should be subject to the access control rule (Bates et al., col. 9, line 62 – col. 10, line 28);

a third control parameter for specifying an expiration time for the access control rule (Bates et al., fig. 7, elem. 217);

and a fourth control parameter for identifying the access control message (Bates et al., fig. 2).

Regarding claim 25, the combination enables:

determining validity of the access control message based on the third control parameter (Bates et al., fig. 3);

Regarding claim 26, the combination enables:

determining executability of the target computer content based on the second control parameter (Bates et al., col. 9, line 62 – col. 10, line 28);

.

Regarding claims 27 and 28, they are rejected for the same reasons as claims 20 and 22, and further because the combination enables the usage of their system in a network of communicating computers (Bates et al., fig. 1). Communications to a user can be blocked when computer content is deemed to be untrustworthy (Bates et al., col. 3, lines 24-27, col. 14, line 6 – col. 15, line 8).

Regarding claim 29, the combination enables:

wherein the data communication is blocked when the target computer content is time-stamped not before the virus alert time (Bates et al., fig. 3; fig 7).

Regarding claim 30, it is rejected, at least, for the same reasons as claim 1, and furthermore because the combination enables:

a firewall module monitoring data communications initiated by a target computer content and sending a request to examine the data communications (Bates et al., fig. 1, elems.20, 30, 50). The combination enables that the system is useful in a network and it is capable of filtering trustworthy and untrustworthy computer content – thus, acting as a firewall module.

an access control console, for generating an access control message specifying an access control rule for blocking data communications of the target executable computer file that has a time stamp corresponding to an earliest moment the computer file was allowed to execute, and the time-stamp is not before a virus alert time, the access control message including a first control parameter for computing the virus alert time in response to a virus outbreak report indicating a virus attack threat to a computer network (Bates et al., fig. 7; fig. 2);

and an access control module, coupled to the access control console and the firewall module, configured to receive the access control message and a request from the firewall module, and to compute the virus alert time based on the virus access control time and to determine whether the data communication should be blocked *based on the access control rule* (Bates et al., fig. 1, elem. 44, see rejections of claims 1 and 2).

Regarding claim 31, it is a program and computer medium claim implementing the method claim 1, and it is rejected for the same reasons (see also, Bates et al., fig. 1).

Regarding claim 32, it is rejected, at least, for the same reasons as claim 1, and furthermore because the combination enables:

means for entering a computer virus status mode in response to a virus outbreak report indicating a virus attack threat to a computer network and for automatically recovering a preselected virus access control time (Bates et al., fig. 7);

coupled to the entering and recovering means, means for computing a virus alert time based on the virus access control time *(Bates et al., fig. 1, elems. 31, 42, 44),* and coupled to the computing virus alert time means, means for comparing a time stamp of a target computer content with the virus alert time prior to execution of the computer content *(Bates et al., fig. 1, elem. 42),*

and for determining the executability of the computer content in response to comparing the time stamp of the target computer content with the virus alert time (Bates et al., col. 9, line 56 – col. 10, line 8; col. 11, lines 12-24). The combination enables a determination of computer content executability is performed for determining the trustworthiness ("executability") of content.

Regarding claim 34, it comprises essentially similar limitations, and it is rejected, at least, for the same reasons as claim 1.

Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of Bates et al. and Hericourt in view of Symantec, "Norton AntiVirus Corporate Edition".

Regarding claim 11, the combination enables that viruses can be found in email attachments, and that it is well known in the art for antivirus programs to have the capability for performing antivirus processing on emails and email attachments (Bates et al., col. 1, lines 35-63). The combination enables an antivirus program or module for performing such antivirus processing (Bates et al., fig. 1, elems. 44, 52). Bates et al., however, does not disclose the details of the antivirus processing for emails and email attachments. Specifically, Bates et al. does not disclose that the antivirus program or module removes the computer content from the E-mail body, and denies execution of the computer content.

Symantec discloses an antivirus program and the details of how the program performs antivirus processing upon an email with an attachment. Symantec discloses that the antivirus program scans content attached to an email body and removes such content if it is found to contain a virus, thus, denying execution of the content (Symantec, page 15, par. 2; page 22, "Managing Realtime Protection").

It would have been obvious for one of ordinary skill in the art to combine the details disclosed by Symantec for the antivirus processing of emails with the system of

Bates et al. because the system of The combination enables an antivirus program capable of performing antivirus processing for processing of emails.

(10) Response to Argument

Appellant's arguments filed 2/13/09 have been fully considered but they are not persuasive.

Appellant argues:

The cited references, at the least, fail to disclose a time stamp "corresponding to an *earliest moment* the computer code was allowed to execute on a computer..." or "corresponding to a *first time* the computer code was allowed to execute on a computer..." ... Thus Bates at most discloses use of a single time stamp indicating the time the file was *last checked* for a virus. There is no teaching or suggestion in Bates of a time stamp that indicates the earliest moment or first time computer code was allowed to execute.

In fact, Bates does not even teach or suggest recording computer code execution times.

(Brief, pg. 8, 9)

Examiner responds:

The examiner has relied upon the prior art teachings that virus scanning comprises code execution, as demonstrated by the combination of Bates and Hericourt. Thus, in response to applicant's arguments against the references individually (i.e. Bates), one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Appellant argues:

Hericourt does not teach or suggest using time stamps for any purpose. (Brief, pg. 9)

Examiner responds:

The examiner has relied upon the prior art teachings that virus scanning comprises code execution and the utilization of time stamps, as demonstrated by the combination of Bates and Hericourt. Thus, in response to applicant's arguments against the references individually (i.e. Hericourt), one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Appellant argues:

The combination of Bates and Hericourt does not render the claimed invention obvious because neither reference discloses or suggests using a time stamp that corresponds to an earliest moment, or first time, the computer code was allowed to execute on a computer coupled to a computer network. (Brief, pg. 9)

Examiner responds:

In response, the examiner respectfully notes that the combination of Bates and Hericourt enables a networked computer, such as a file download site, to virus scan (i.e. execute) and timestamp an executable, the timestamp corresponds to the time when the file was scanned/executed by the networked computer. This virus scanned file is provided to thousands of users who are enabled to subsequently execute this file upon their client computers coupled to the network (Bates, fig. 1:20, 50; col. 4, lines 35-55; col. 8, lines 12-15, 43-48).

Thus, the examiner points out that the facts clearly reveal a sequence of operations enabled by the networked computer system (Bates, fig. 1) of the prior art. First, a file is virus scanned/executed by a networked computer. Second, the timestamp representing the scanning/execution of the file is added to a memory table. Third, the file is provided to up to thousands of users, wherein it is subsequently executed upon client computers.

The examiner respectfully notes that the appellant appears to recognize this sequence. In fact, regarding the sequence of first scanning/executing a file by a networked computer and then downloading and executing the file by a multitude of client computers, the examiner respectfully notes that appellant's admission:

"*...there are an arbitrary number of executions that occur ...* ***after the execution represented by the timestamp***". (Applicant's Arguments/Remarks, 4/11/08, pg. 14)

Therefore, it is respectfully pointed out that if the applicant acknowledges a system wherein a sequence of software executions occur, wherein the sequence may consist of a time-stamped execution (i.e. virus scanning of the software) and a plurality of subsequent executions (i.e. execution of the software by thousands of client computers), then the appellant must also reasonably acknowledge that the time-stamped execution represents ***"an earliest"*** or "***a first***" time of execution in relation to the multitude of subsequent, secondary, or later executions within that sequence.

Appellant argues:

The combination does not teach or suggest time stamping the earliest or first time code was executed; there might have been prior executions occurring before the time-stamped execution. Moreover, even if a particular time stamp in Bates did represent the earliest code execution, this happenstance event would not support the rejection. See In re Robertson, 169 F.3d 743,745 (Fed. Cir. 1999) (stating that the mere fact that a certain thing may result from a given set of circumstances is not sufficient to support a rejection based on inherency); MPEP 2112 IV. Thus, a person of ordinary skill in the art at the time the invention was made, considering the teachings of the references either alone or in combination, would not find the claimed invention obvious. (Brief, pg. 10)

Examiner responds:

In response, the examiner respectfully notes that the claims do not recite *"time stamping the earliest or first time code was executed".*

Instead, the claim language more explicitly states that the timestamp corresponds to "***an*** *earliest moment* the computer code was allowed to execute on a computer coupled to the computer network" and "***a*** *first time* the computer code was allowed to execute on a computer coupled to the computer network".

As noted within the rejection, the examiner points out that the recited *"an earliest moment the computer code was allowed to execute on a computer coupled to the computer network"* must be given its broadest reasonable interpretation in harmony with the appellant's specification. Within the appellant's specification, the appellant reveals that this claimed timestamp of *"an earliest moment the computer code was allowed to execute"* represents a time when an executable file is virus checked by a networked computer and the resulting timestamp and hash value from the virus check is added to a database or memory table (e.g. see appellant's specification, par. 38, 50, 51, 60, 61; original claims 21 and 33).

Accordingly, the examiner notes that the prior art combination enables a *"computer coupled to the computer network"* (e.g. file hosting site [50], search engine [30] – see Bates, fig. 1; col. 8, lines 12-15, 43-48) to virus scan an executable, the resulting timestamp of virus scanned code corresponding to an "execution time" since virus scanning comprises execution of the executable (Hericourt, col. 3, lines 25-54). This scanning/execution of the executable by the networked computer causes the virus status information (e.g. the timestamp) to be added to a database or memory table (Bates, col. 8, lines 12-15, 43-48; col. 13, lines 24-34). Thus, the networked computer safely provides these virus-scanned, executable files as downloads, whereby they are subsequently executed at later times by thousands of users with client computers (e.g. see Bates, fig. 1: 20, 30, 50; col. 4, lines 35-55).

Because the prior art discloses a sequence of executions, beginning with a timestamped virus scanning and a multitude of secondary and later executions, the examiner notes that the prior art combination clearly enables a timestamp corresponding to "***an** earliest"* or *"**a** first" "moment the computer code was allowed to execute on a computer coupled to the computer network"* (Hericourt, col. 3, lines 25-54; Bates, fig. 1: 50, 30, 20; Bates col. 8, lines 12-15, 43-48).

Appellant argues:

The Examiner's defense of this rejection is essentially predicated on an unreasonable interpretation of the claims. Claims must be given their broadest reasonable interpretation consistent with the specification. Phillips v. A WH Corp., 415 F.3d 1303 (Fed. Cir. 2005); MPEP § 2111. Here, the Examiner's interpretation is unreasonable because it calls code executions "earliest" or "first" even if prior code executions have occurred on a computer coupled to the network. (Brief, pg. 10

In other words, the Examiner calls any given time-stamped execution the "earliest" or "first" one by arbitrarily excluding from consideration any executions that came before it. For example, if there were three sequential executions, the Examiner would call execution two "an earliest" execution because it occurred before execution three, even though execution one occurred earlier. Moreover, if there were a sequence of 10 time stamps, the Examiner would call nine of the time stamps "first." This interpretation is improper because it is clearly unreasonable. (Brief, pg. 11)

Examiner responds:

In response, the examiner respectfully notes that the appellant appears to mischaracterize the examiner's stated rejection and arguments. Specifically, the examiner does not call any arbitrary execution "the first" or "the earliest" code execution. Instead, the examiner is simply noting the logic that a code execution (such as the virus scanning of an executable by a file server) can rightly be considered to be "**a** first" or "**an** earliest" ***in relation*** *to* all other subsequent, secondary, later code executions (e.g. the multitude of executions of the virus scanned file by a plurality of client computers) .

The examiner notes that it is not unreasonable to consider a beginning of an identified sequence of events (e.g. code "executions") to be "a first" or "an earliest" in comparison to all subsequent events within the identified sequence.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Jeffery Williams/

Examiner, Art Unit 2437

Conferees:

/Emmanuel L. Moise/

Supervisory Patent Examiner, Art Unit 2437

/Matthew B Smithers/
Primary Examiner, Art Unit 2437